

Image encoding method and image encoding/decoding method.

Patent Number: EP0598995, B1
Publication date: 1994-06-01
Inventor(s): OKA KENICHIRO C O MITSUBISHI D (JP)
Applicant(s): MITSUBISHI ELECTRIC CORP (JP)
Requested Patent: JP6164950
Application Number: EP19930113923 19930831
Priority Number(s): JP19920316759 19921126
IPC Classification: H04N1/41
EC Classification: H04N1/41C4B, G06T9/00S
Equivalents: DE69314908D, DE69314908T, KR123790
Cited Documents: EP0261561; EP0549309; JP1188166

Abstract

An encoding method of a GBTC type that encoded data and decoded data have a fixed length, which can reduce the degree of deterioration of the quality of a reproduced image and also to provide an encoding/decoding method which, in encoding or decoding, is capable of performing editing processings to rotate an original image in 90 DEG and to rotate it inversely with respect to up and down, right and left. When encoded data in each of blocks of an original image and level specification signals ϕ_{ij} of respective pixels in each block are written into a memory or are read out therefrom, the fractions of parameter values are rounded to the nearest whole number or cut away to thereby put operation values into integral numbers, or the encoded data and the pixel level specification signal are arranged in such a manner that they can be rotated or inversely rotated an integral number of times 90 DEG . 

Data supplied from the esp@cenet database - I2

3 (4) 基準レベル L_a は、ブロック間差分を可変長符号で符号化すること。

【0005】このようにブロック毎に符号化された可変長符号は、各ブロック内に符号化された長さが異なる。従って画像圧縮のために符号データ系列の中から任意のブロックを取り出すことは困難である。さらに、基準レベル L_a はブロック間差分の符号化である。さらに、基準レベル L_a はブロック部分の符号だけを取り出しても復号化できず、符号化時と同じ順序で、最初のブロックから順番に復号していくなければならない。

【0006】このような問題点を解決する符号化方法として、[画像電子学会研究会予稿、「ハードコピーバージョンにおける画像圧縮方式の評価」1991-04-01]

10 (3) 基準レベル L_a は、ブロック間差分を可変長符号で符号化すること。

【0007】図5は、このGBT形符号化方法がある。以下に説明する。図5は、このGBT形符号化方法の説明図である。図5において、1は原画像、2は 4×4 画素の正方形で構成されるブロック、 X_{ij} ($i, j = 1, 2, 3, 4$) は画素データ ($i, j = 1, 2, 3, 4$) における「 X_{ij} 」という)、 ϕ_{ij} ($i, j = 1, 2, 3, 4$) はレベル指定信号 ($i, j = 1, 2, 3, 4$) という)である。

【0008】表1に、従来の符号化データの固定量化した結果を符号化方式のアルゴリズムを示す。

【0009】

【表1】

GBT形符号化アルゴリズム	
$P_1 = (L_{max} + 3L_{min})/4$ $P_2 = (3L_{max} + L_{min})/4$ $Q_1 = \text{mean of all } \phi_{ij} \text{ such that } X_{ij} \leq P_1$ $Q_2 = \text{mean of all } \phi_{ij} \text{ such that } X_{ij} > P_2$ $L_d = (Q_1 + Q_2)/2$ $L_d = Q_4 - Q_1$ $L_1 = L_d - L_d/4$ $L_2 = L_d + L_d/4$ $\text{for } i = 1, \dots, 4$ $\quad \text{for } j = 1, \dots, 4$ $\quad \quad \text{if } X_{ij} \leq L_1 \quad \phi_{ij} = 01 \text{ (binary)}$ $\quad \quad \text{else if } X_{ij} \leq L_2 \quad \phi_{ij} = 10 \text{ (binary)}$ $\quad \quad \text{else if } X_{ij} \leq L_2 \quad \phi_{ij} = 10 \text{ (binary)}$ $\quad \quad \text{else} \quad \phi_{ij} = 11 \text{ (binary)}$ $\quad \text{end_if}$ end_for end_for	
復号化アルゴリズム	
$\text{for } i = 1, \dots, 4$ $\quad \text{for } j = 1, \dots, 4$ $\quad \quad \text{if } \phi_{ij} = 01 \quad y_{ij} = L_a - L_d/2$ $\quad \quad \text{else if } \phi_{ij} = 00 \quad y_{ij} = L_a - L_d/6$ $\quad \quad \text{else if } \phi_{ij} = 10 \quad y_{ij} = L_a + L_d/6$ $\quad \quad \text{else} \quad y_{ij} = L_a + L_d/2$ $\quad \text{end_if}$ end_for end_for	

【0010】図6は、この符号化方法を用いた信号処理系のブロック回路図で、3はホスト、4は符号回路、5は復号化回路、6は画像データ用のメモリ (以下「メモリ」という) である。

【0011】また、図7はこの符号化回路4のブロック回路図で、7はバッファメモリ、8は符号化演算回路、9は基準レベルバッファ、10はレベル制御バッファ、11はレベル指定信号バッファ、12は信号制御回路である。

【0012】つぎに、符号化方法について説明する。この説明例では、画素データ X_{ij} は各1ビットで構成さ

れていますと仮定する。原画像1を 4×4 画素毎のブロック2に分割し、ブロック毎に符号化する。すなわち、まず、ブロック2の内の画素の最大値 L_{max} と最小値 L_{min} の間を4等分し、下から4分の1の値を P_1 、上から4分の1の値を P_2 とする。つぎに、 L_{min} 以上 P_1 以下の画素の平均値を Q_1 、 L_{max} 以下 P_2 より大きい画素の平均値を Q_2 とする。つぎにブロックの基準レベル L_a を

$L_a = (Q_1 + Q_2) / 2$
 $L_d = Q_2 - Q_1$

7

8

として求める。つぎに、レベル間隔 L_d を 4 等分し、下から 4 分の 1 の値を L_1 、上から 4 分の 1 の値を L_2 とする。

*

つぎに、ブロック内の 16 画素をそれぞれ L_1 、 L_2 を閾値として下記のように 4 領に量子化する。

$L_{min} \leq x_{ij} \leq L_1$ の場合 $\phi_{ij} = 01$ (2 進)

$L_1 < x_{ij} \leq L_2$ の場合 $\phi_{ij} = 00$ (2 進)

$L_2 < x_{ij} \leq L_{max}$ の場合 $\phi_{ij} = 10$ (2 進)

$\phi_{ij} = 11$ (2 進)

【0013】つぎに、符号化回路 4 の動作を、図 7において、画像データは図 8 に示すように、符号データは図 8 について説明する。図 7において、画像データはいたんバッファメモリ 7 に 4 ライン分をされ、プロ

セク毎に符号化演算回路 8 に送られ、そして、既に符号化された符号データは、 L_a 、 L_d 、 $\phi_{ij} = 1 \sim 4$ が求められ、それを基にレベル間隔バッファ 9、レベル間隔バッファ 10、レベル指定データ 11 を経て符号化回路 12 に送られ、図 8 (a) に示すブロック内の符号データとしてまとめられる。

【0014】この符号データは図 8 (b) のように配置された符号系列としてメモリ 6 に書き込まれる。 L_a と L_d にそれぞれ 1 バイト、 $\phi_{ij} = 1 \sim 4$ に 2 ビットづつ割り当たると、1 ブロック分の符号データ量は 6 バイトになる。このように、各ブロックの符号量は 6 バイト※ 算を用いて符号化データ y_{ij} を求める。

$\phi_{ij} = 01$ の場合 $y_{ij} = L_a - L_d / 2$

$\phi_{ij} = 00$ の場合 $y_{ij} = L_a - L_d / 6$

$\phi_{ij} = 10$ の場合 $y_{ij} = L_a + L_d / 6$

$\phi_{ij} = 11$ の場合 $y_{ij} = L_a + L_d / 2$

【0018】つぎに、符号化回路 5 の動作を説明する。図 8において、1 ブロック分の符号データは符号化演算回路 4 で L_a 、 L_d 、 $\phi_{ij} = 1 \sim 4$ に分けられ、この 3 種類の符号はそれぞれ基準レベルバッファ 15、レベル間隔バッファ 16、レベル指定符号バッファ 17 に送り分けられ、符号化演算回路 18 で符号化データ y_{ij} が計算され、復元された y_{ij} はブロック毎にバッファメモリ 19 に書き込まれ、4 ライン毎にに出力される。

【0019】[発明が解決しようとする課題] 従来の符号長を固定化した GBT C 形符号化方法は、符号化時の各パラメータ P_1 、 P_2 、 L_a 、 L_d 、 L_1 、 L_2 を求める演算、および符号化時の符号化データ y_{ij} を求める演算において符号化処理が生じるため、同一の画像データに対して符号化処理を繰り返すよなが用いた場合には、処理を重ねる度に符号化データや符号化データが変化し続ける、画像として再生したときに画質が劣化し続けるという問題点があった。

【0020】この発明は、上記のような問題点を解決するためになされたもので、符号量を固定化した GBT C 形符号化方法において、符号化と復元化を繰り返した場合も再生画質が原画からの劣化が小さい範囲に留まる符號化方法を得ることを目的とする。

【0021】また、符号化時、または復元化時に、画像を 90° 単位で回転させる機能または上下、左右を反転

* それぞれ 3 回目以内の符号あるいは復号化で収束する。

【0026】また、この発明に係る画像符号化/復号化方法は、符号化時に、符号データおよび各ブロック内のレベル指定信号を 90° の整数倍の回転または反転操作を含めて配置するようにして復号化時の画像腐食が可能になる。

【実施例 1】実施例 1、表 2 に、この発明に係る GBT C 形符号化方式のアルゴリズムを示す。

【0029】

[表 2]

(6) 特開平6-164950

10

* 整倍の回数または反転操作を含めて配置して復号化するようにして復号化時の画像腐食が可能になる。

【0028】

【実施例 1】実施例 1、表 2 に、この発明に係る GBT C 形符号化方式のアルゴリズムを示す。

[表 2]

実施例 1のGBT C 形符号化アルゴリズム

```

P1 = (lmax + 3lmin) / 4      (四捨五入又は切り捨て)
P2 = (3lmax + lmin) / 4      (四捨五入又は切り捨て)
Q1 = mean of all xij such that xij < P1 (四捨五入又は切り捨て)
Q4 = mean of all xij such that xij ≥ P2 (四捨五入又は切り捨て)
La = (Q1 + Q4) / 2      (四捨五入又は切り捨て)
Ld = Q4 - Q1      (四捨五入又は切り捨て)
L1 = La - Ld / 4      (四捨五入又は切り捨て)
L2 = La + Ld / 4      (四捨五入又は切り捨て)
for(i = 1, ..., 4)
    for(j = 1, ..., 4)
        if xij ≤ L1
            φij = 01 (binary)
        else if xij > La
            φij = 00 (binary)
        else if xij ≤ L2
            φij = 10 (binary)
        else
            φij = 11 (binary)
    end_if
end_for
end_for

```

復号化アルゴリズム

```

for(i = 1, ..., 4)
    for(j = 1, ..., 4)
        if φij = 01
            yij = La - Ld / 2 (四捨五入又は切り捨て)
        elseif φij = 00
            yij = La - Ld / 6 (四捨五入又は切り捨て)
        elseif φij = 10
            yij = La + Ld / 6 (四捨五入又は切り捨て)
        else
            yij = La + Ld / 2 (四捨五入又は切り捨て)
    end_if
end_for
end_for

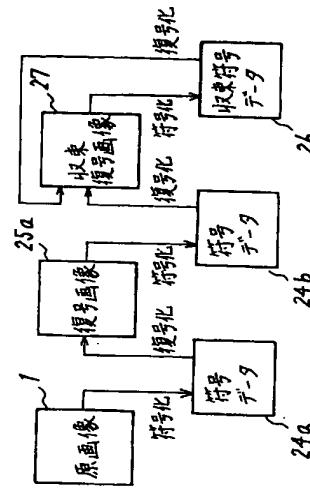
```

【0030】この符号化方式のアルゴリズムと、表 1 に示した既来の符号化方式のアルゴリズムの相違点は、 Q_{ij} を求める演算量に端数が生じたとき、四捨五入、4 を $1lmax$ 以下、 P_2 以上の演算量の平均値とするこ

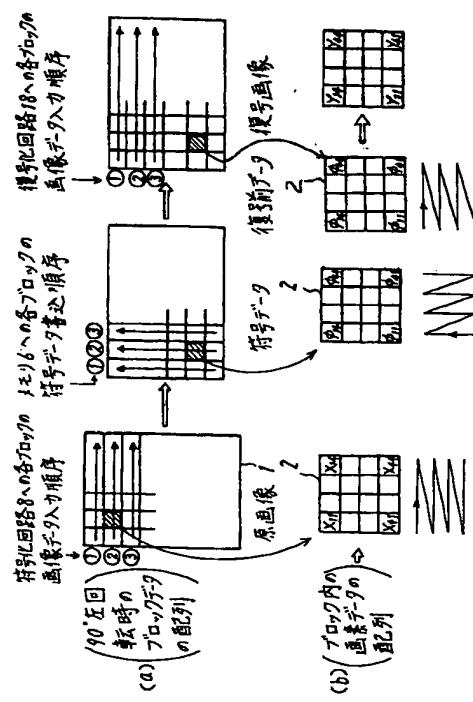
と、パラメータ P_1 、 P_2 、 Q_1 、 Q_4 、 L_a 、および

または切り捨てる整数化処理を施すことであって、この

21

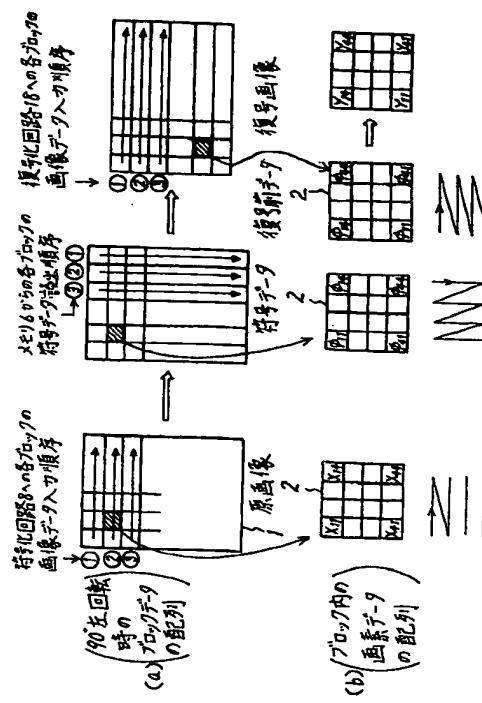


13



(10)

41



四三

2

音声データ

音声認識

音声回路

音声解釈

音声合成

算出回路

音声出力

音声データ

音声認識

音声回路

音声解釈

音声合成

算出回路

音声出力

音声データ

